CS316 Project 3: Design & Implementation of a TCP File Service

This project is to be done in teams of two.

Learning objectives:

1. Implement the server-client architecture.
2. Design and implement a file transfer protocol.
3. Apply the TCP byte stream model to implement reliable data transfer.

What to submit:

1. The java source files.
2. The five flowcharts for the five commands (see an example on the next page)
    a. In each flowchart, also include a diagram to illustrate the format and content of the client request message.

Each team will just need to submit *one* copy. **Please be sure to add a submission note stating the names of both team members**.

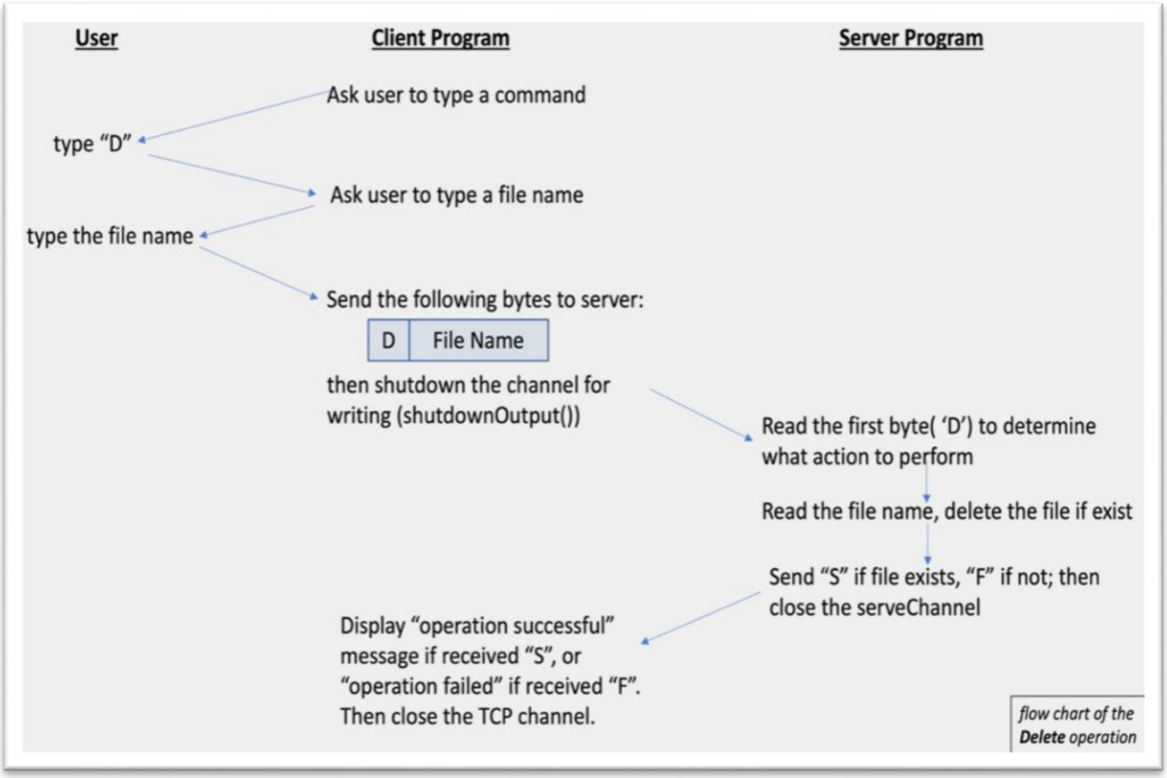
Instructions:

1. The file **client** should provide five commands to users, as follows:
    a. *List*: let user see a list of files available on the server
    b. *Delete*: let user delete a file on the server
    c. *Rename*: let user rename a file on the server
    d. *Download*: let user download a file from the server
    e. *Upload*: let user upload a file to the server.

   Note that the user only operates the client program, and not the server.

   See below for a flowchart for the “delete” command, as an example. Note that here the letter “D” is used to code the “delete” command and letters “F” and “S” are used as the server’s status codes. This is just an example and you may code the commands/replies differently.

   For “delete”, “rename”, and “upload” commands, the server should sent a status code to the client, to indicate whether the operation has been completed successfully. A status code is not required for the “list” or “download” operations.



Ask user to type a command

type "D"

Ask user to type a file name

type the file name

Send the following bytes to server:

then shutdown the channel for writing (shutdownOutput())

Read the first byte('D') to determine what action to perform

Read the file name, delete the file if exist

Send "S" if file exists, "F" if not; then close the serveChannel

Display "operation successful" message if received "S", or "operation failed" if received "F". Then close the TCP channel.

flow chart of the ***Delete*** *operation*

2. The **client** program requires two command-line arguments to run: the first one is the IP address of the server (if the server runs on the same machine as the client, use *localhost*), and the second one is the port number the server runs on.
3. Always run the server program first before running the client program -- the client will try to connect to the server, so make sure the server is ready to accept incoming TCP connection requests.
4. The final testing needs to be done with the server and the client running on two different computers.

Grading:

This project carries 100 points, as follows:

- Each of the five commands carries 15 points (for a total of 75 points).
  For *each* command:
  - Client correctly sends out the request and receives the reply: 6 points.
  - Server correctly receives & processes the request and sends out the reply: 6 points.
  - The implementation of the client & server matches the flowchart: 3 points.
- The remaining 25 points are for the correct implementation of the overall structure of the client and the server, including the correct use of conditional and/or switch statements, reading from keyboard, handling invalid commands, etc.

You will receive partial credit if your implementation is partially correct.